**Lesson on Ternary Operators in C**

**Introduction**

The ternary operator is a compact and elegant way to write conditional statements in C. It allows you to make decisions and assign values in a single line of code, making your code more concise and, in many cases, easier to read.

**What is a Ternary Operator?**

The ternary operator is the only operator in C that takes three operands. It is used as a shorthand for if-else statements. The general syntax of the ternary operator is:

condition ? expression\_if\_true : expression\_if\_false;

* **condition**: This is the condition that will be evaluated. If it's true, the first expression (expression\_if\_true) is executed. If it's false, the second expression (expression\_if\_false) is executed.
* **expression\_if\_true**: The expression that runs if the condition is true.
* **expression\_if\_false**: The expression that runs if the condition is false.

**Example Usage**

Let's look at a basic example:

#include <stdio.h>

int main() {

int a = 10;

int b = 20;

// Using ternary operator to find the maximum value

int max = (a > b) ? a : b;

printf("The maximum value is %d\n", max);

return 0;

}

**Explanation**:

* In this example, the ternary operator checks if a is greater than b.
* If a > b is true, max is assigned the value of a.
* If a > b is false, max is assigned the value of b.

**When to Use the Ternary Operator**

* **Simple Conditions**: The ternary operator is ideal for simple conditions where using an if-else statement would feel cumbersome and unnecessarily verbose.
* **Assignment in a Single Line**: If you want to assign a value based on a condition in a single line, the ternary operator is the perfect tool.

**Practical Exercise**

**Objective**: Write a C program that uses the ternary operator to determine whether a given number is even or odd.

**Exercise Steps**

1. **Prompt the user** to enter an integer.
2. **Use the ternary operator** to check if the number is even or odd.
3. **Print the result** to the screen.

**Example Program**:

#include <stdio.h>

int main() {

int number;

// Prompt user for input

printf("Enter an integer: ");

scanf("%d", &number);

// Use ternary operator to check if the number is even or odd

char \*result = (number % 2 == 0) ? "even" : "odd";

// Print the result

printf("The number %d is %s.\n", number, result);

return 0;

}

**Expected Outcome**

When the user runs this program:

* If they input 4, the program should output: "The number 4 is even."
* If they input 7, the program should output: "The number 7 is odd."

**Extended Challenge**

Modify the program to check if the input number is positive, negative, or zero, using the ternary operator.

**Hint**: You will need to use nested ternary operators.

#include <stdio.h>

int main() {

int number;

// Prompt user for input

printf("Enter an integer: ");

scanf("%d", &number);

// Use nested ternary operator to check if the number is positive, negative, or zero

char \*result = (number > 0) ? "positive" : (number < 0) ? "negative" : "zero";

// Print the result

printf("The number %d is %s.\n", number, result);

return 0;

}

**Expected Outcome for the Extended Challenge**

* If the user inputs 5, the program should output: "The number 5 is positive."
* If the user inputs -3, the program should output: "The number -3 is negative."
* If the user inputs 0, the program should output: "The number 0 is zero."

**Conclusion**

Ternary operators are a powerful tool in C for simplifying conditional logic, making your code more concise and often more readable. While they’re perfect for simple conditions, remember that clarity is key—avoid using them for complex conditions where an if-else statement would be more understandable.

**Further Reading and Practice**

* Try converting simple if-else statements in your code to ternary operators.
* Practice using nested ternary operators, but be cautious of readability.
* Explore how other programming languages implement or differ in their use of ternary operators.